Titanic Data Analysis Project

Ashley Kemuma

2025-07-31

1.Installation and data loading

Load Libraries After installation of the necessary packages, confirm if the packages are available for use in your R session.

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
library(tidyr)
```

Load the data and display the first five rows

```r
df<- read.csv("/Users/ashleykemuma/Desktop/Titanic Data Analysis Project/week9_titanic_eda_data.csv")
head(df)
```

```
##   PassengerID Survived Pclass   Name    Sex  Age SibSp Parch Ticket   Fare
## 1           1        0      3 Name_1 female 35.6     2     1  65207  71.42
## 2           2        1      3 Name_2   male 19.5     3     3  67643  57.08
## 3           3        0      3 Name_3 female 26.1     3     4  50593 152.16
## 4           4        0      3 Name_4 female 10.4     3     0  57762 124.40
## 5           5        0      3 Name_5 female 28.5     1     4  26461 117.36
## 6           6        1      1 Name_6 female 45.6     0     4  41574  38.37
##   Cabin Embarked
## 1  C123        C
## 2   D33        Q
## 3              Q
## 4    G6        Q
## 5   D33        C
## 6    G6        C
```

2.Data Inspection *Summarry of the Dataset*

dim(df): Returns the number of rows and columns.

str(df): Provides a concise summary of the data frame's structure, including variable names, data types, and a few observations.

summary(df): Generates descriptive statistics (min, max, mean, median, quartiles) for numeric variables and counts for categorical variables.

colSums(is.na(df)): Calculates the total number of missing values for each column, which is vital for planning data cleaning steps.

```r
dim(df)
```

```
## [1] 5000   12
```

```r
str(df)
```

```
## 'data.frame':    5000 obs. of  12 variables:
##  $ PassengerID: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 0 0 0 1 0 0 0 1 ...
##  $ Pclass     : int  3 3 3 3 3 1 3 3 1 1 ...
##  $ Name       : chr  "Name_1" "Name_2" "Name_3" "Name_4" ...
##  $ Sex        : chr  "female" "male" "female" "female" ...
##  $ Age        : num  35.6 19.5 26.1 10.4 28.5 45.6 39.6 49.6 25.8 48.8 ...
##  $ SibSp      : int  2 3 3 3 1 0 0 2 3 3 ...
##  $ Parch      : int  1 3 4 0 4 4 1 3 4 4 ...
##  $ Ticket     : int  65207 67643 50593 57762 26461 41574 62842 68180 80279 20465 ...
##  $ Fare       : num  71.4 57.1 152.2 124.4 117.4 ...
##  $ Cabin      : chr  "C123" "D33" "" "G6" ...
##  $ Embarked   : chr  "C" "Q" "Q" "Q" ...
```

```r
summary(df)
```

```
##   PassengerID      Survived          Pclass          Name          
##  Min.   :   1   Min.   :0.0000   Min.   :1.000   Length:5000       
##  1st Qu.:1251   1st Qu.:0.0000   1st Qu.:1.000   Class :character  
##  Median :2500   Median :0.0000   Median :2.000   Mode  :character  
##  Mean   :2500   Mean   :0.4992   Mean   :2.001                     
##  3rd Qu.:3750   3rd Qu.:1.0000   3rd Qu.:3.000                     
##  Max.   :5000   Max.   :1.0000   Max.   :3.000                     
##                                                                    
##      Sex                 Age              SibSp           Parch      
##  Length:5000        Min.   :-24.90   Min.   :0.000   Min.   :0.000  
##  Class :character   1st Qu.: 20.57   1st Qu.:1.000   1st Qu.:1.000  
##  Mode  :character   Median : 29.80   Median :2.000   Median :3.000  
##                     Mean   : 29.94   Mean   :2.462   Mean   :2.564  
##                     3rd Qu.: 39.40   3rd Qu.:4.000   3rd Qu.:4.000  
##                     Max.   : 79.40   Max.   :5.000   Max.   :5.000  
##                     NA's   :300                                     
##      Ticket           Fare            Cabin             Embarked        
##  Min.   :10021   Min.   : 10.04   Length:5000        Length:5000       
##  1st Qu.:32547   1st Qu.: 58.34   Class :character   Class :character  
##  Median :54537   Median :105.56   Mode  :character   Mode  :character  
##  Mean   :54803   Mean   :105.85                                        
##  3rd Qu.:77298   3rd Qu.:154.06                                        
##  Max.   :99998   Max.   :199.98                                        
## 
```

```r
colSums(is.na(df))
```

```
## PassengerID    Survived      Pclass        Name         Sex         Age 
##           0           0           0           0           0         300 
##       SibSp       Parch      Ticket        Fare       Cabin    Embarked 
##           0           0           0           0           0           0
```

3.Data Cleaning and Preparation This phase addresses common data quality issues, specifically focusing on missing values, and creates new features that can enhance the analysis. Missing values are imputed to ensure that all observations can be used in subsequent steps, and new categorical variables are derived for richer insights. *3.1 Handle Missing Age Values* Missing Age values are imputed using the median age of all passengers. The median is chosen over the mean because age distributions can often be skewed by outliers, making the median a more robust measure of central tendency.

```r
# Impute missing Age values with the median
df$Age[is.na(df$Age)] <- median(df$Age, na.rm = TRUE)
# Verify no more missing Age values
sum(is.na(df$Age))
```

```
## [1] 0
```

3.2 Handle Missing Embarked Values Missing values in the Embarked column (port of embarkation) are imputed with the mode (the most frequently occurring port). This is a common strategy for categorical missing data.

```r
# Find the mode of Embarked
mode_embarked <- names(sort(table(df$Embarked), decreasing = TRUE))[1]
# Impute missing Embarked values
df$Embarked[is.na(df$Embarked)] <- mode_embarked
# Verify no more missing Embarked values
sum(is.na(df$Embarked))
```

```
## [1] 0
```

3.3 Create Age_Group A new categorical variable, Age_Group, is created by binning the continuous Age variable into "Youth", "Adult", and "Senior" categories. This allows for analysis of survival patterns across different life stages.

```r
df$Age_Group <- cut(df$Age,
                    breaks = c(0, 25, 40, Inf),
                    labels = c("Youth (18-25)", "Adult (26-40)", "Senior (41+)"),
                    right = TRUE, include.lowest = TRUE)
# Check the distribution of the new age groups
table(df$Age_Group)
```

```
## 
## Youth (18-25) Adult (26-40)  Senior (41+) 
##          1656          2171          1101
```

3.4 Create Family_Size A Family_Size variable is calculated by summing SibSp (number of siblings/spouses aboard) and Parch (number of parents/children aboard), and adding 1 for the passenger themselves. This helps understand the impact of family presence on survival.

```r
df$Family_Size <- df$SibSp + df$Parch + 1
# Check the distribution of family size
table(df$Family_Size)
```

```
## 
##   1   2   3   4   5   6   7   8   9  10  11 
## 142 283 406 527 689 830 710 579 404 297 133
```

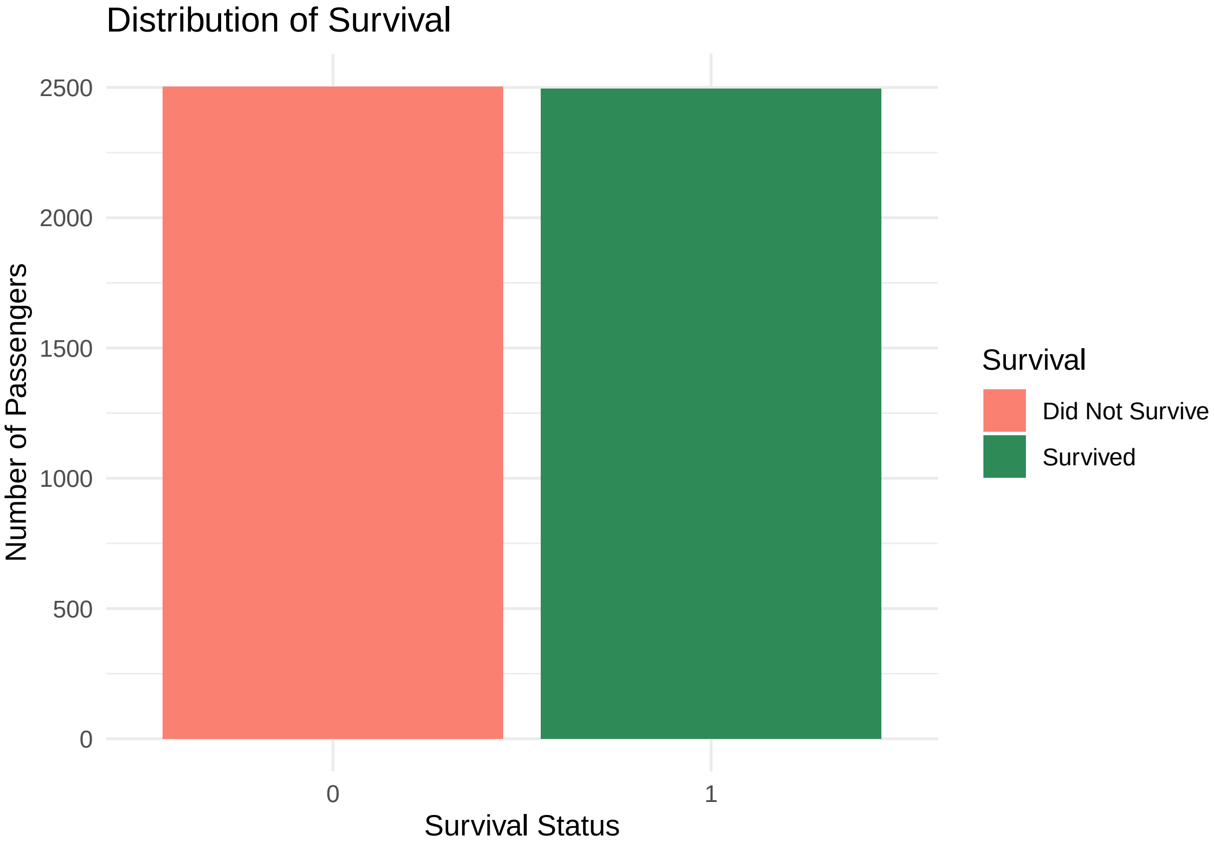
3.5 Convert Categorical Variables to Factors Key categorical variables are converted to R's factor data type. This is crucial for ggplot2 to treat them as discrete categories for plotting and for many statistical models to interpret them correctly. Value labels are implicitly handled by ggplot2 when factors are used.

```
df$Survived <- as.factor(df$Survived) # 0 = Did Not Survive, 1 = Survived
df$Pclass <- as.factor(df$Pclass)     # 1 = 1st, 2 = 2nd, 3 = 3rd class
df$Sex <- as.factor(df$Sex)
df$Embarked <- as.factor(df$Embarked)
```

4.Exploratory Data Analysis (EDA) and Visualization with ggplot2 and corrplot This section generates various plots to visually explore the distributions of individual variables and the relationships between them, particularly focusing on how different attributes relate to survival outcomes. ggplot2 is used for most plots due to its flexibility and aesthetic quality, while corrplot is used for the correlation matrix.

4.1 Distribution of Survival (Bar Plot) This bar plot visualizes the count of passengers who survived versus those who did not, providing an immediate overview of the survival outcome.

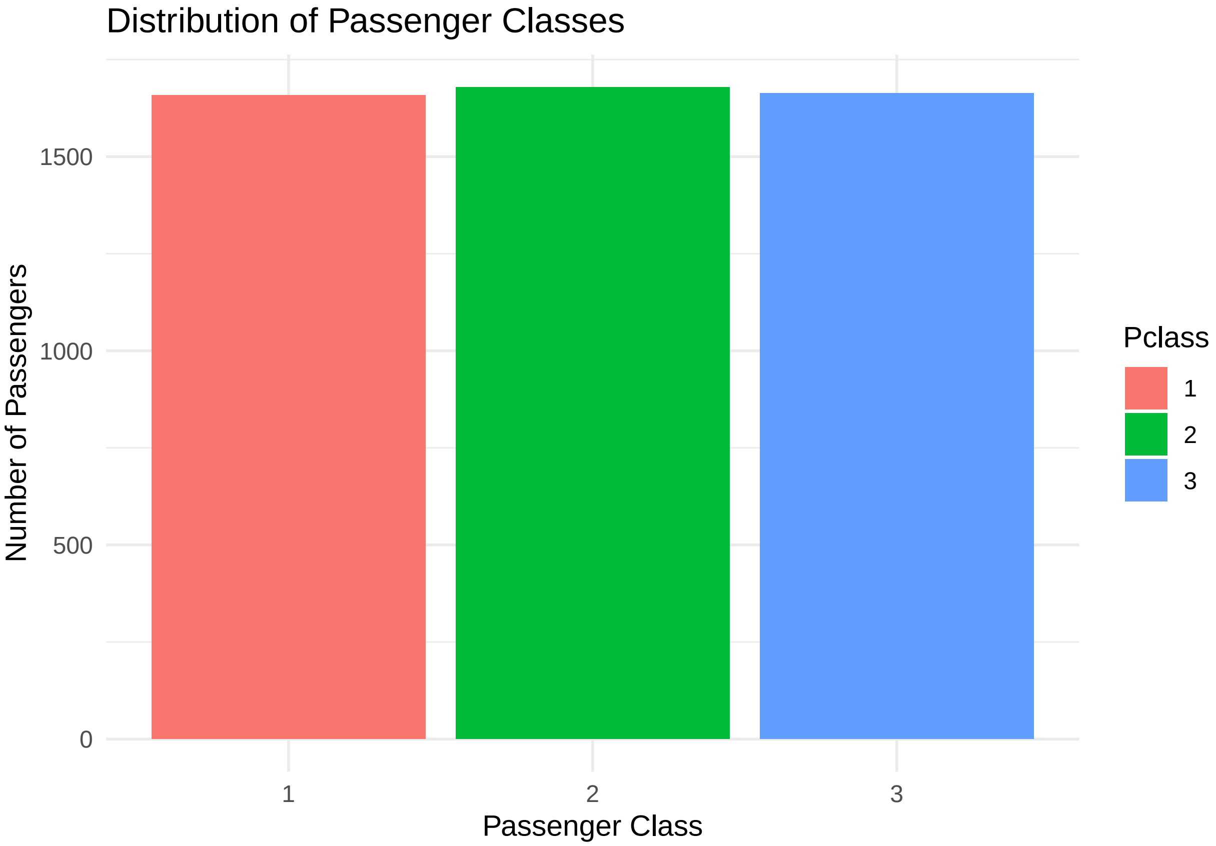
```
ggplot(df, aes(x = Survived, fill = Survived)) +
  geom_bar() +
  scale_fill_manual(values = c("0" = "salmon", "1" = "seagreen"),
                    labels = c("Did Not Survive", "Survived")) +
  labs(title = "Distribution of Survival",
       x = "Survival Status",
       y = "Number of Passengers",
       fill = "Survival") +
  theme_minimal()
```



4.2 Distribution of Passenger Classes (Bar Plot & Pie Chart) These plots show the proportion of passengers in each of the three passenger classes (1st, 2nd, 3rd). This helps understand the composition of the passenger manifest.

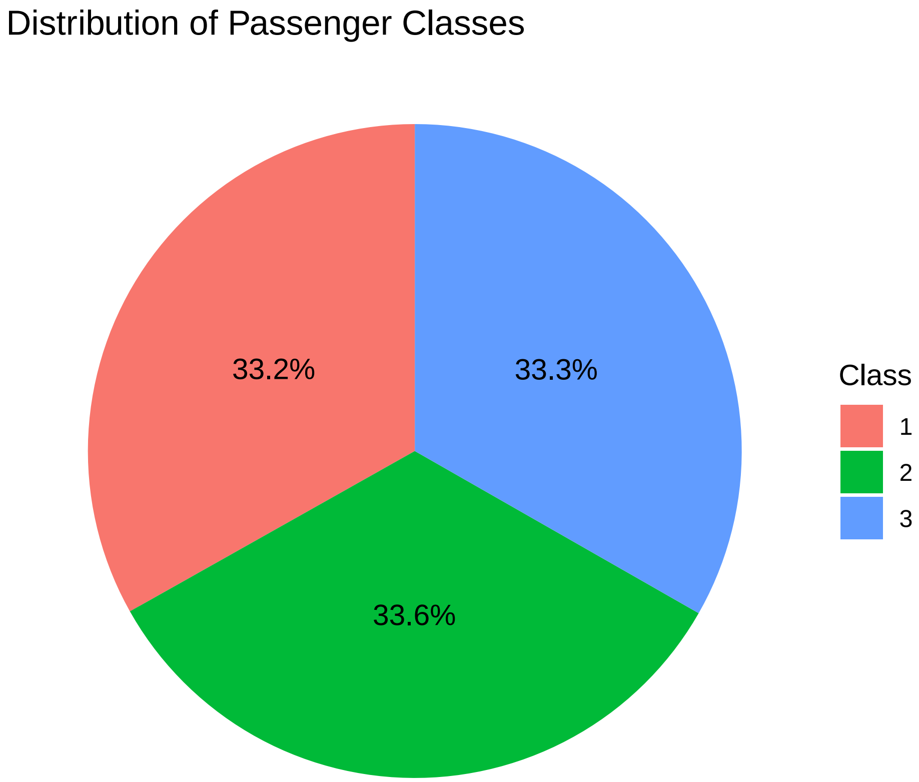
4.2.1 Bar plot

```r
ggplot(df, aes(x = Pclass, fill = Pclass)) +
  geom_bar() +
  labs(title = "Distribution of Passenger Classes",
       x = "Passenger Class",
       y = "Number of Passengers") +
  theme_minimal()
```



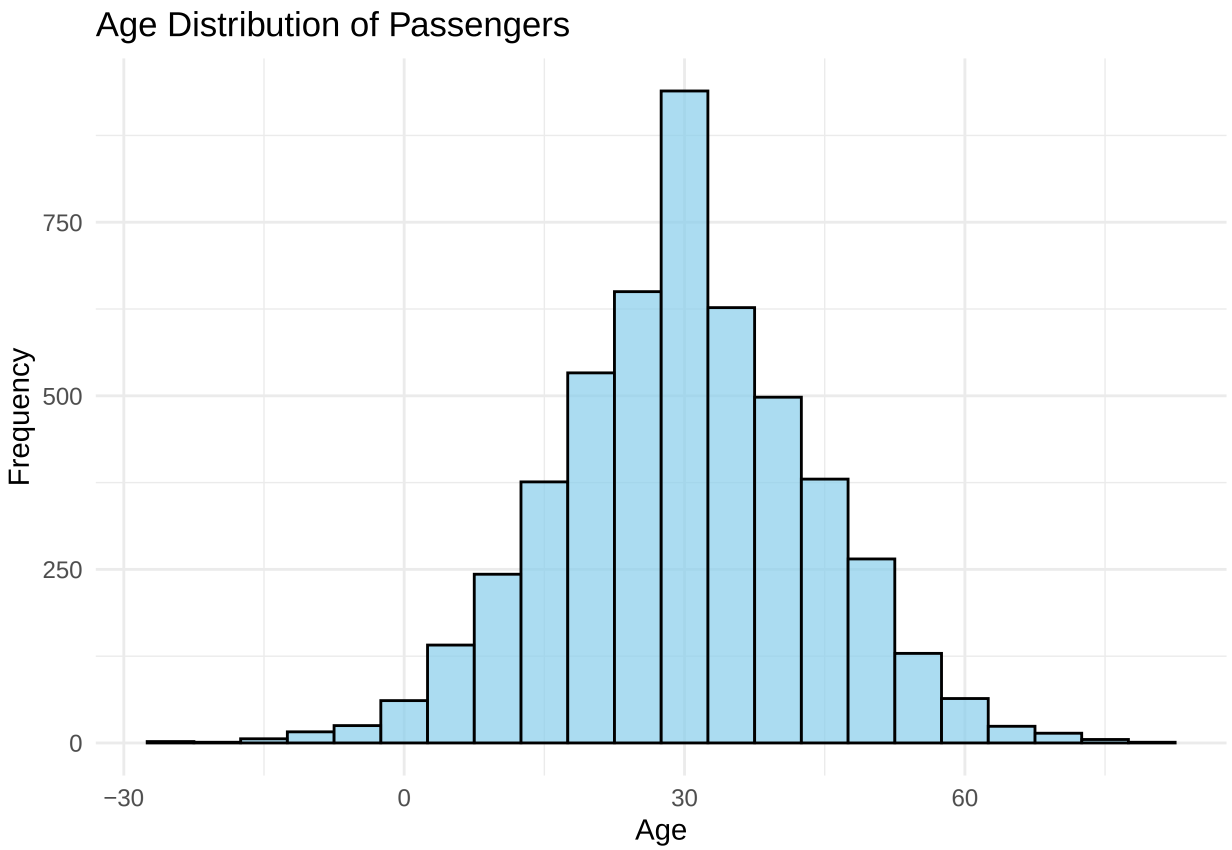
4.2.2 Pie chart

```r
pclass_counts <- table(df$Pclass)
pclass_df <- as.data.frame(pclass_counts)
colnames(pclass_df) <- c("Class", "Count")

ggplot(pclass_df, aes(x = "", y = Count, fill = Class)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Distribution of Passenger Classes", fill = "Class") +
  theme_void() +
  geom_text(aes(label = paste0(round(Count/sum(Count)*100, 1), "%")),
            position = position_stack(vjust = 0.5))
```



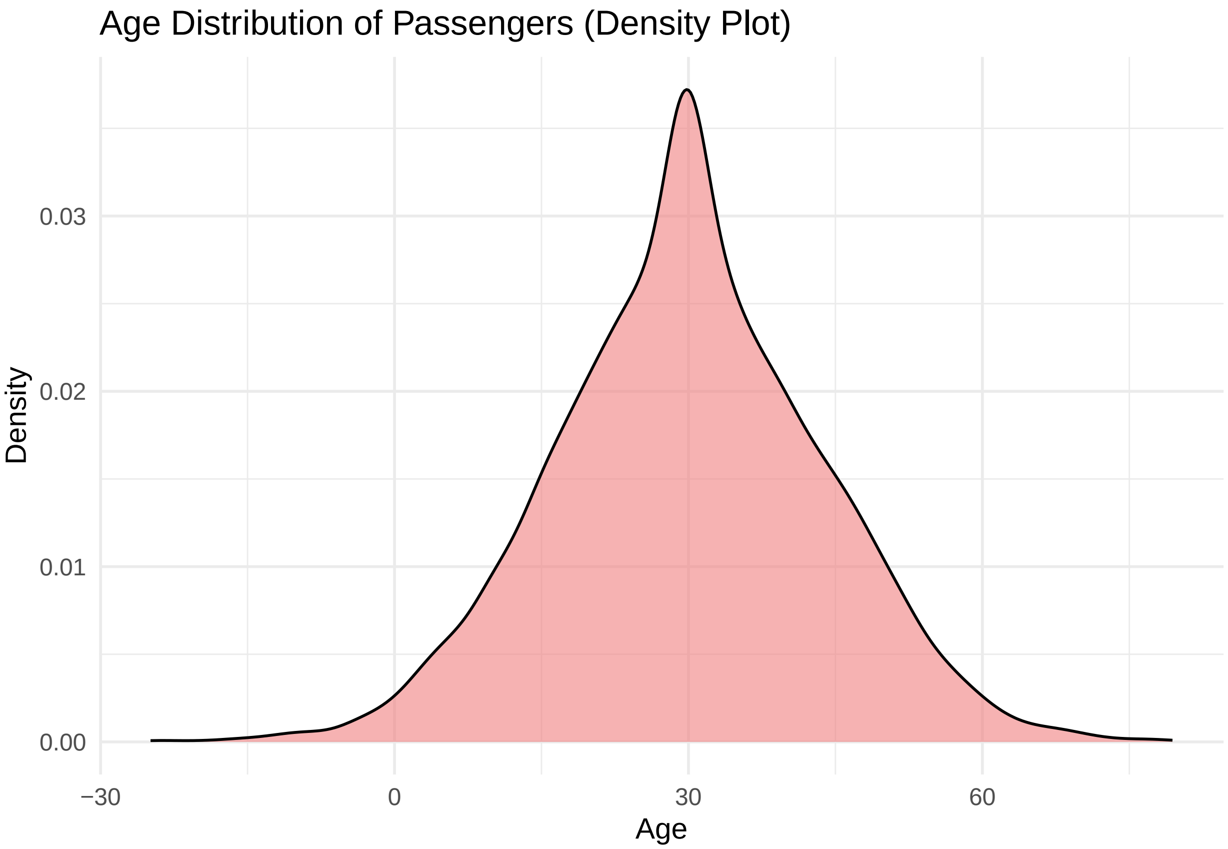
4.3 Age Distribution (Histogram/Density Plot) These plots illustrate the age profile of the passengers. Histograms show frequency bins, while density plots provide a smoothed representation of the distribution. *4.3.1 Histogram*

```r
ggplot(df, aes(x = Age)) +
  geom_histogram(binwidth = 5, fill = "skyblue", color = "black", alpha = 0.7) +
  labs(title = "Age Distribution of Passengers",
       x = "Age",
       y = "Frequency") +
  theme_minimal()
```



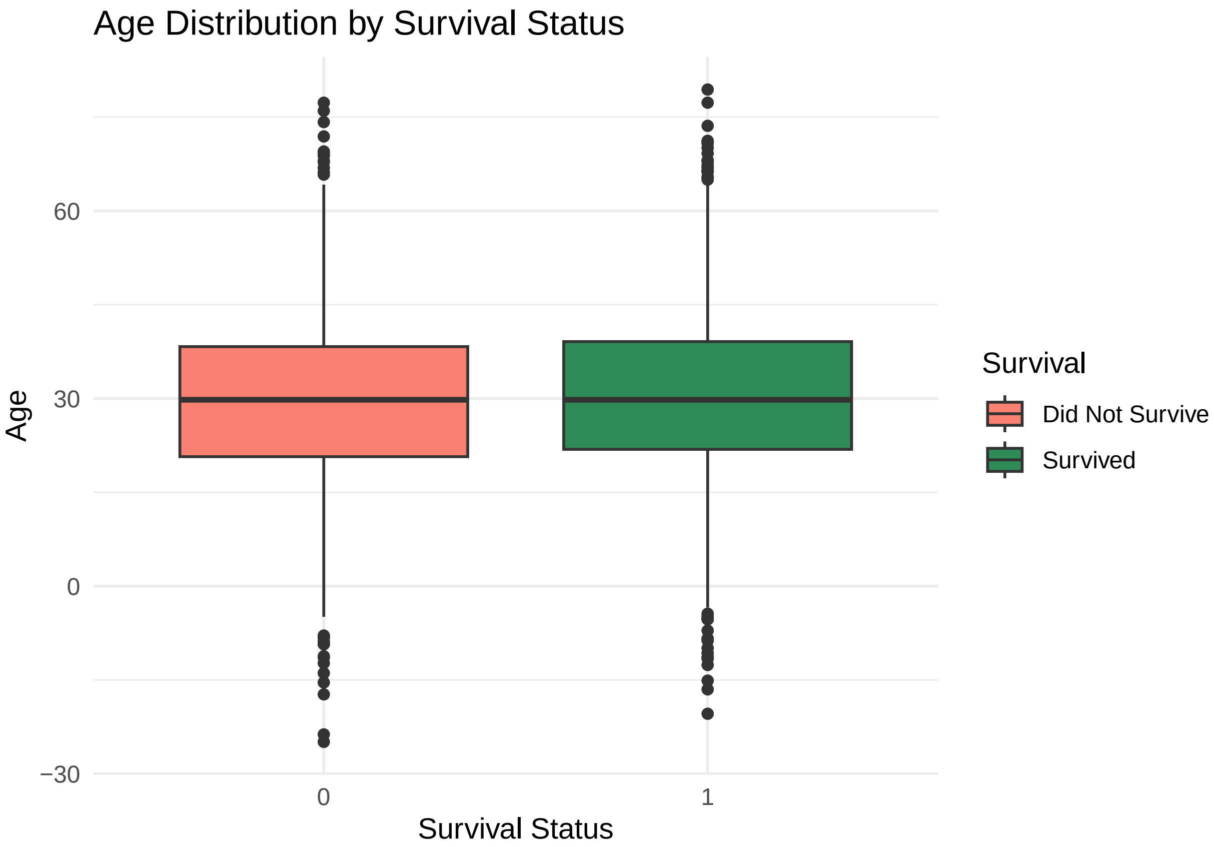
4.3.2 Density plot

```r
ggplot(df, aes(x = Age)) +
  geom_density(fill = "lightcoral", alpha = 0.6) +
  labs(title = "Age Distribution of Passengers (Density Plot)",
       x = "Age",
       y = "Density") +
  theme_minimal()
```

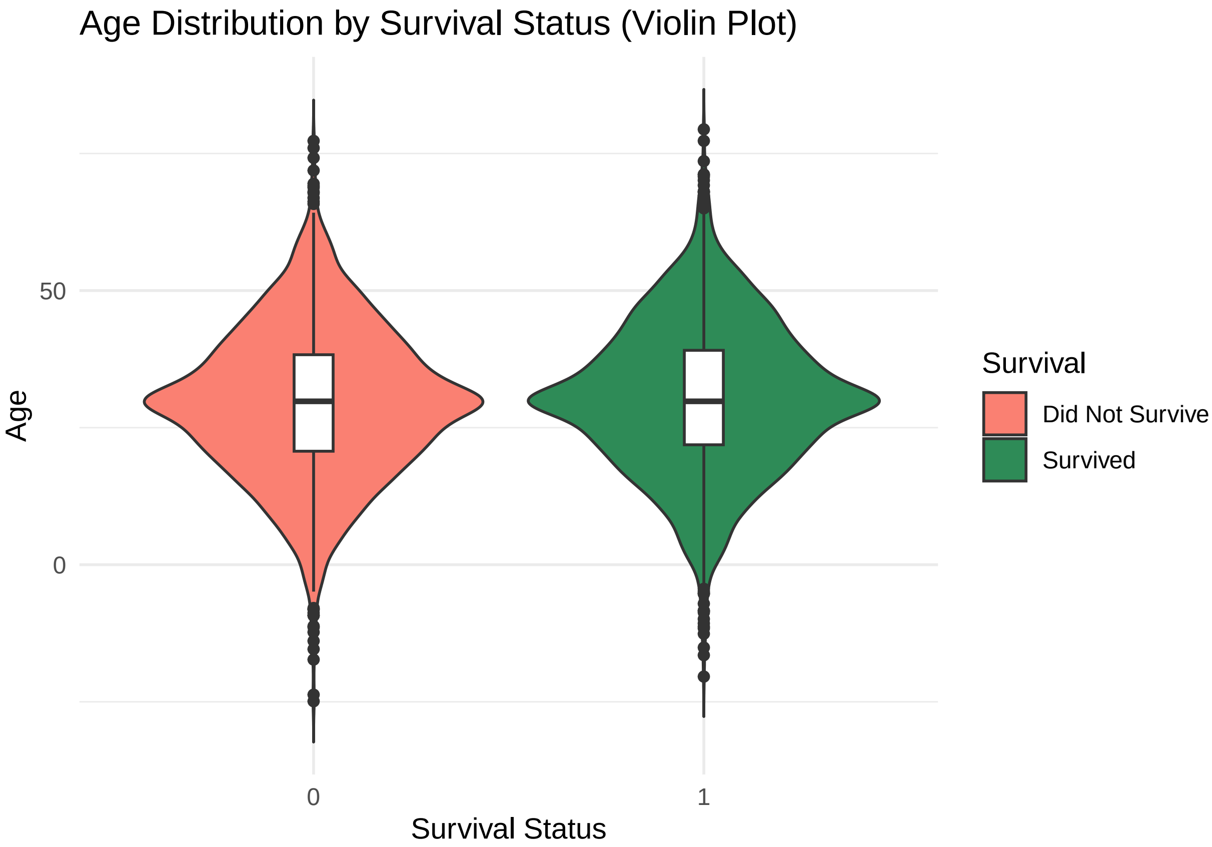


4.4 Relationship between Age and Survival (Boxplots/Violin Plots) These plots compare the age distributions of survivors versus non-survivors. Boxplots show quartiles and outliers, while violin plots additionally show the density of age values at different survival outcomes. *4.4.1 Boxplot*

```r
ggplot(df, aes(x = Survived, y = Age, fill = Survived)) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "salmon", "1" = "seagreen"),
                    labels = c("Did Not Survive", "Survived")) +
  labs(title = "Age Distribution by Survival Status",
       x = "Survival Status",
       y = "Age",
       fill = "Survival") +
  theme_minimal()
```



```r
ggplot(df, aes(x = Survived, y = Age, fill = Survived)) +
  geom_violin(trim = FALSE) +
  geom_boxplot(width = 0.1, fill = "white") + # Add small boxplot inside violin
  scale_fill_manual(values = c("0" = "salmon", "1" = "seagreen"),
                    labels = c("Did Not Survive", "Survived")) +
  labs(title = "Age Distribution by Survival Status (Violin Plot)",
       x = "Survival Status",
       y = "Age",
       fill = "Survival") +
  theme_minimal()
```

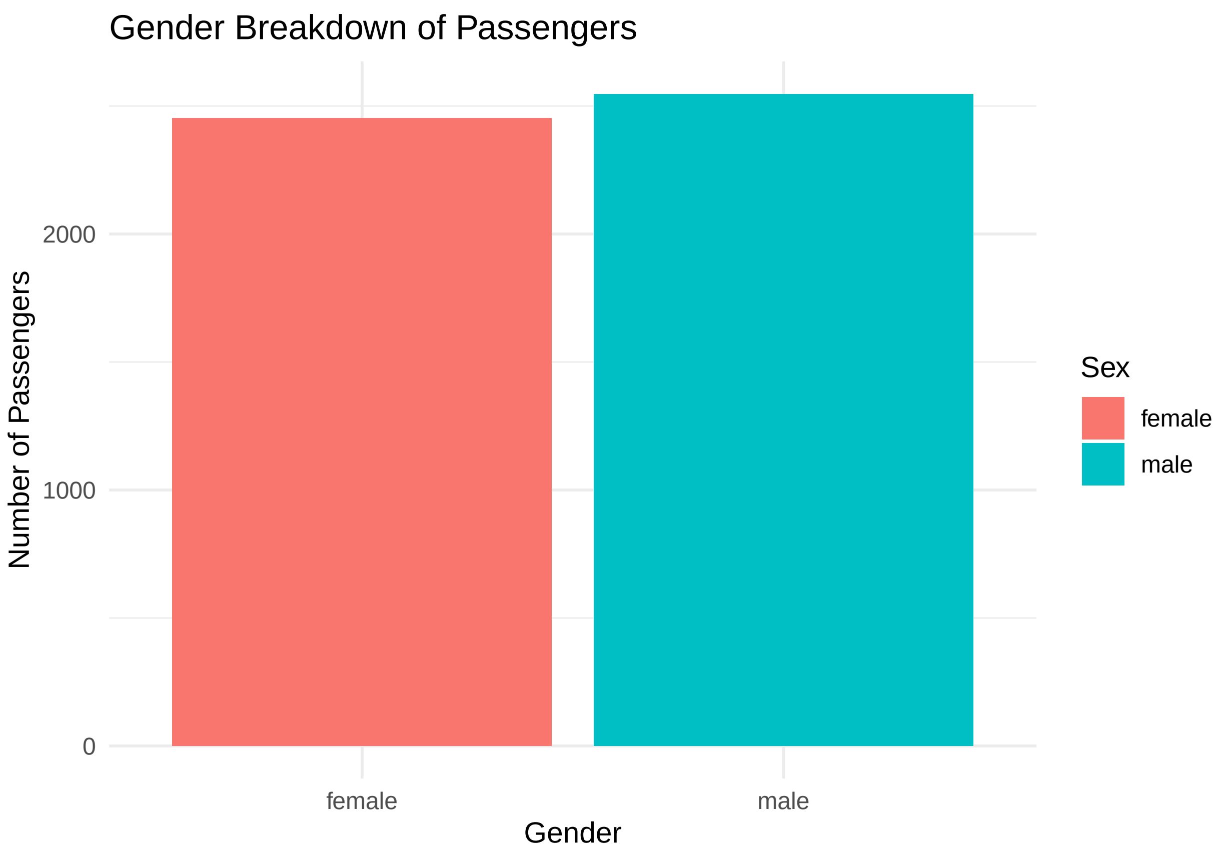


4.5

Gender Breakdown and Survival by Gender (Grouped Bar Chart) These plots show the proportion of male and female passengers and then visualize survival counts broken down by gender, highlighting any disparities.

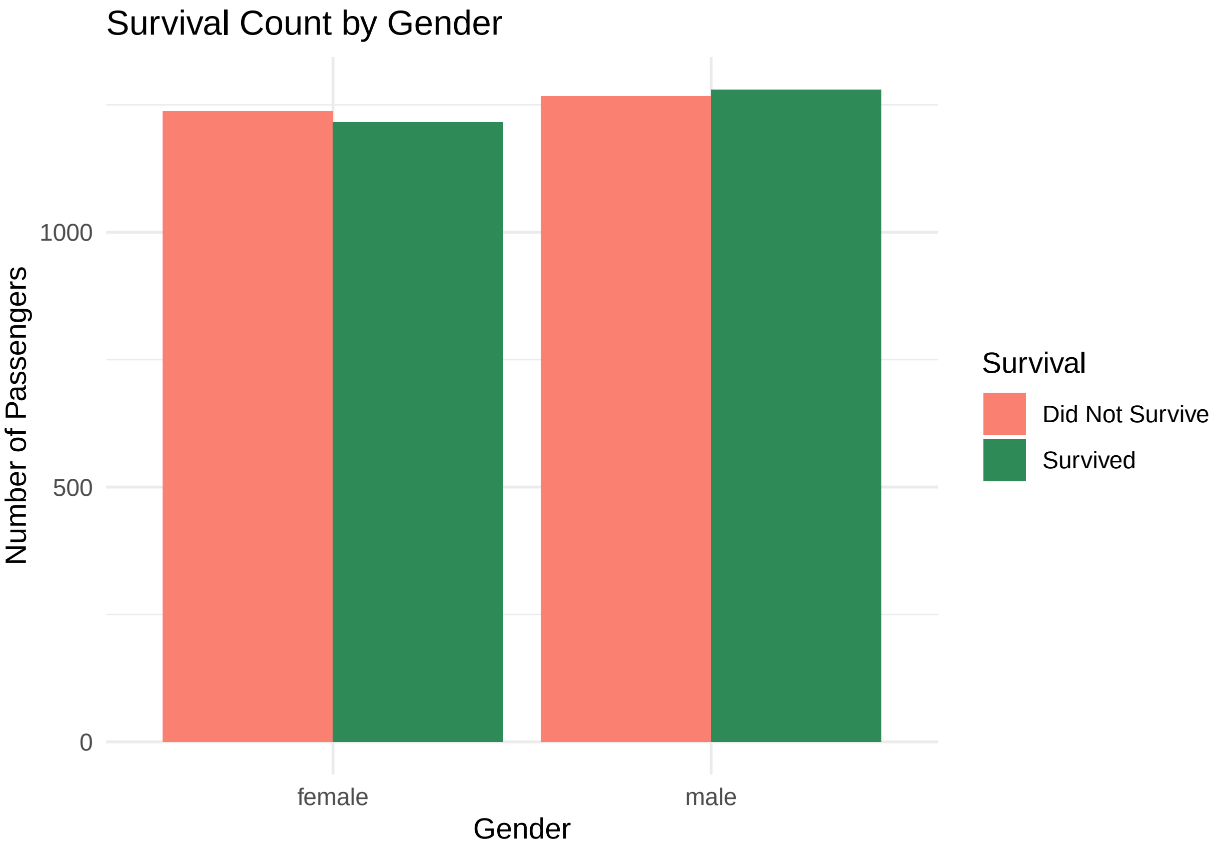
4.5.1 Gender Breakdown

```r
ggplot(df, aes(x = Sex, fill = Sex)) +
  geom_bar() +
  labs(title = "Gender Breakdown of Passengers",
       x = "Gender",
       y = "Number of Passengers") +
  theme_minimal()
```



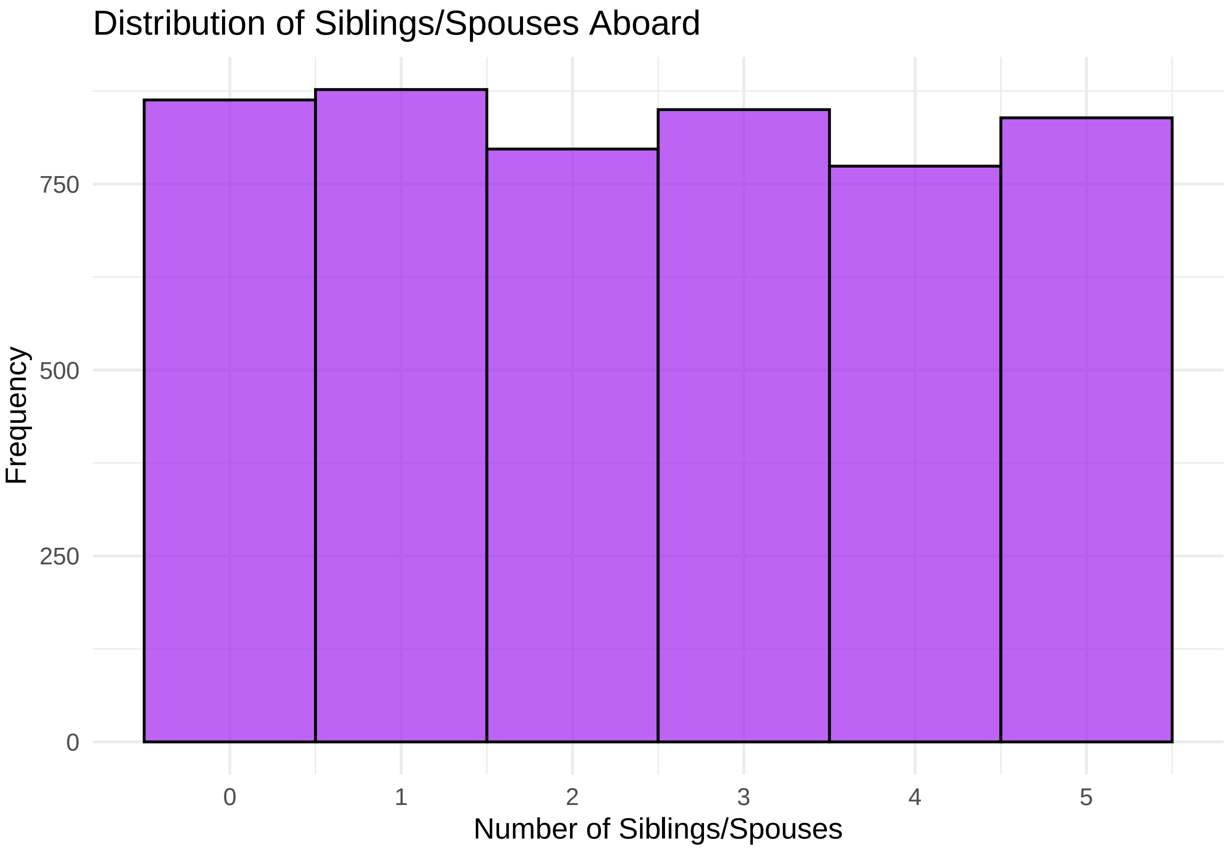
4.5.2 Survival by gender

```
ggplot(df, aes(x = Sex, fill = Survived)) +
  geom_bar(position = "dodge") + # "dodge" for side-by-side bars
  scale_fill_manual(values = c("0" = "salmon", "1" = "seagreen"),
                    labels = c("Did Not Survive", "Survived")) +
  labs(title = "Survival Count by Gender",
       x = "Gender",
       y = "Number of Passengers",
       fill = "Survival") +
  theme_minimal()
```



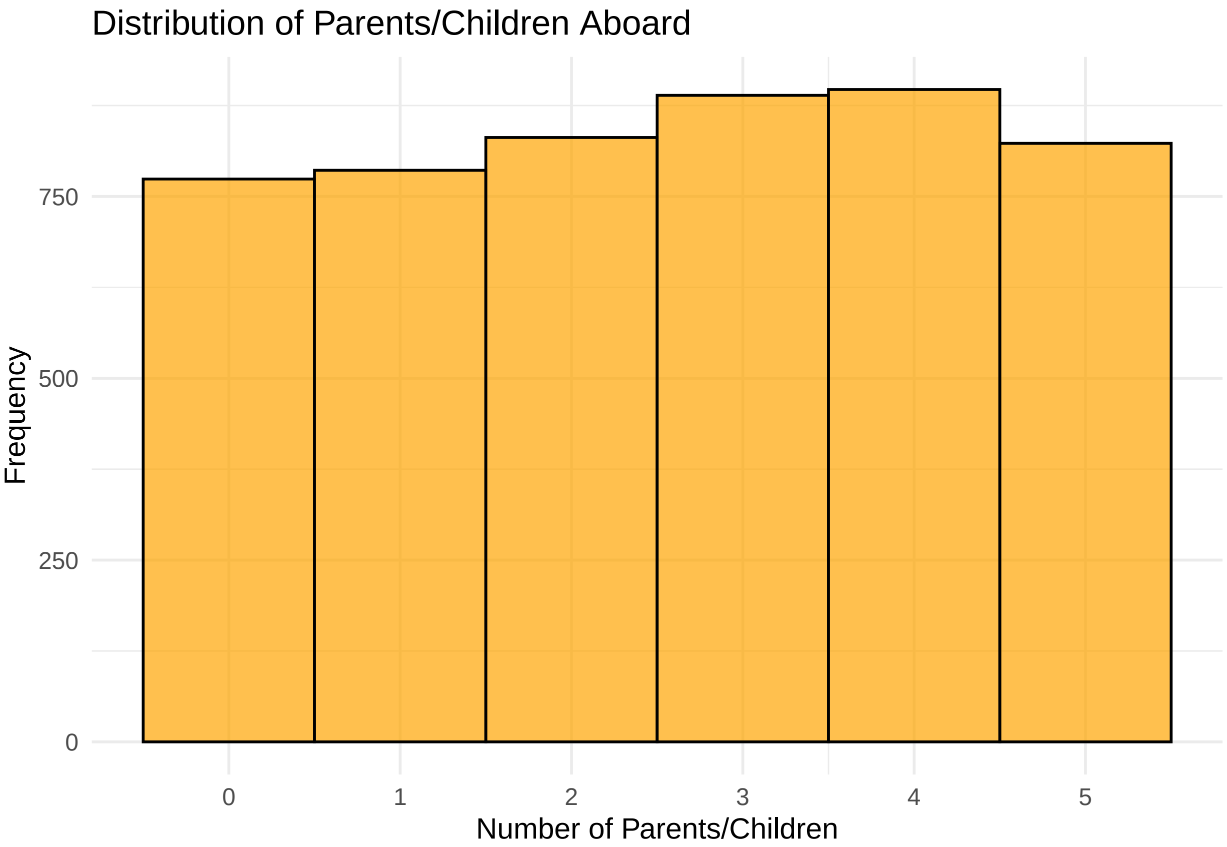
4.6 Siblings/Spouses (SibSp) and Parents/Children (Parch) Distributions These histograms illustrate the distribution of family members (siblings/spouses and parents/children) traveling with each passenger, which can be an important factor in survival. *4.6.1 SibSp Distribution*

```r
ggplot(df, aes(x = SibSp)) +
  geom_histogram(binwidth = 1, fill = "purple", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Siblings/Spouses Aboard",
       x = "Number of Siblings/Spouses",
       y = "Frequency") +
  theme_minimal() +
  scale_x_continuous(breaks = unique(df$SibSp)) # Ensure all integer breaks are shown
```



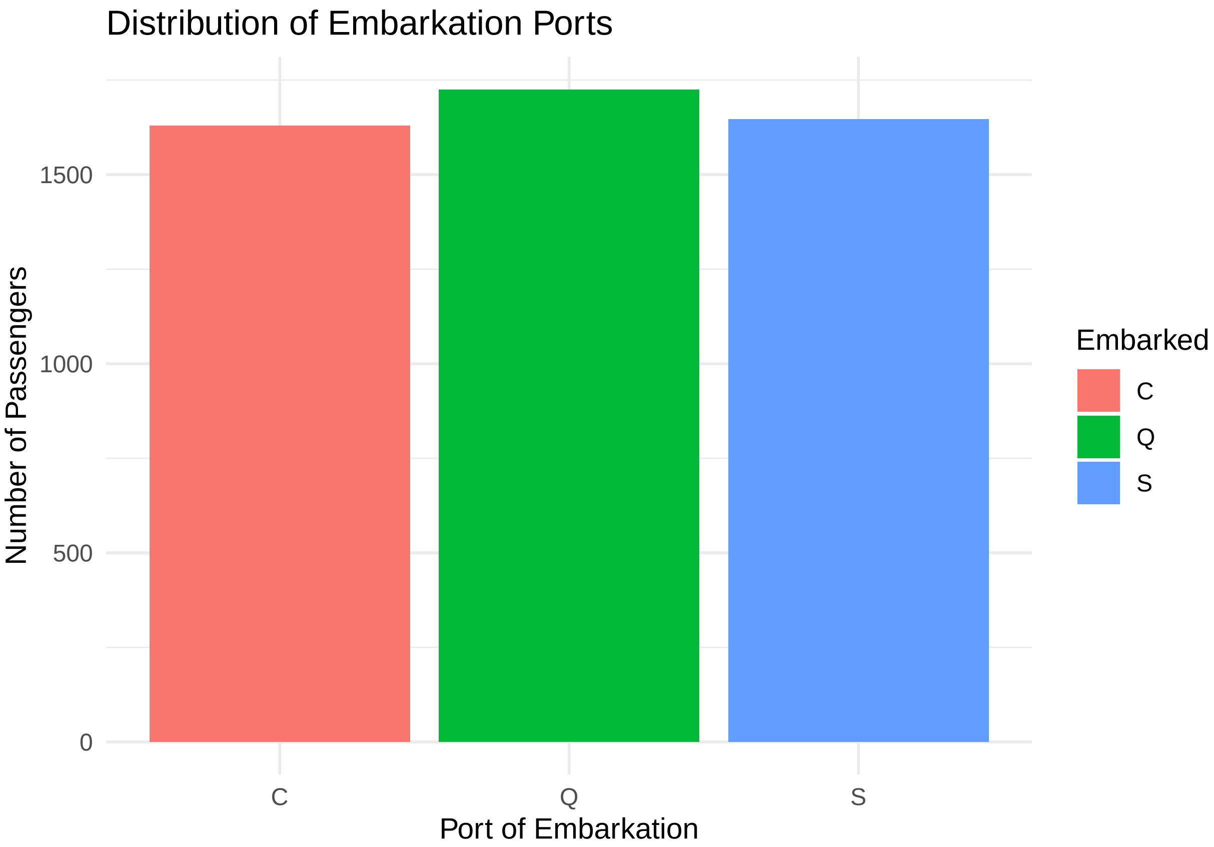
4.6.2 Parch Distribution

```r
ggplot(df, aes(x = Parch)) +
  geom_histogram(binwidth = 1, fill = "orange", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Parents/Children Aboard",
       x = "Number of Parents/Children",
       y = "Frequency") +
  theme_minimal() +
  scale_x_continuous(breaks = unique(df$Parch)) # Ensure all integer breaks are shown
```



4.7 Most Common Embarkation Ports This bar plot shows from which ports passengers embarked, revealing the origin distribution of the passengers.

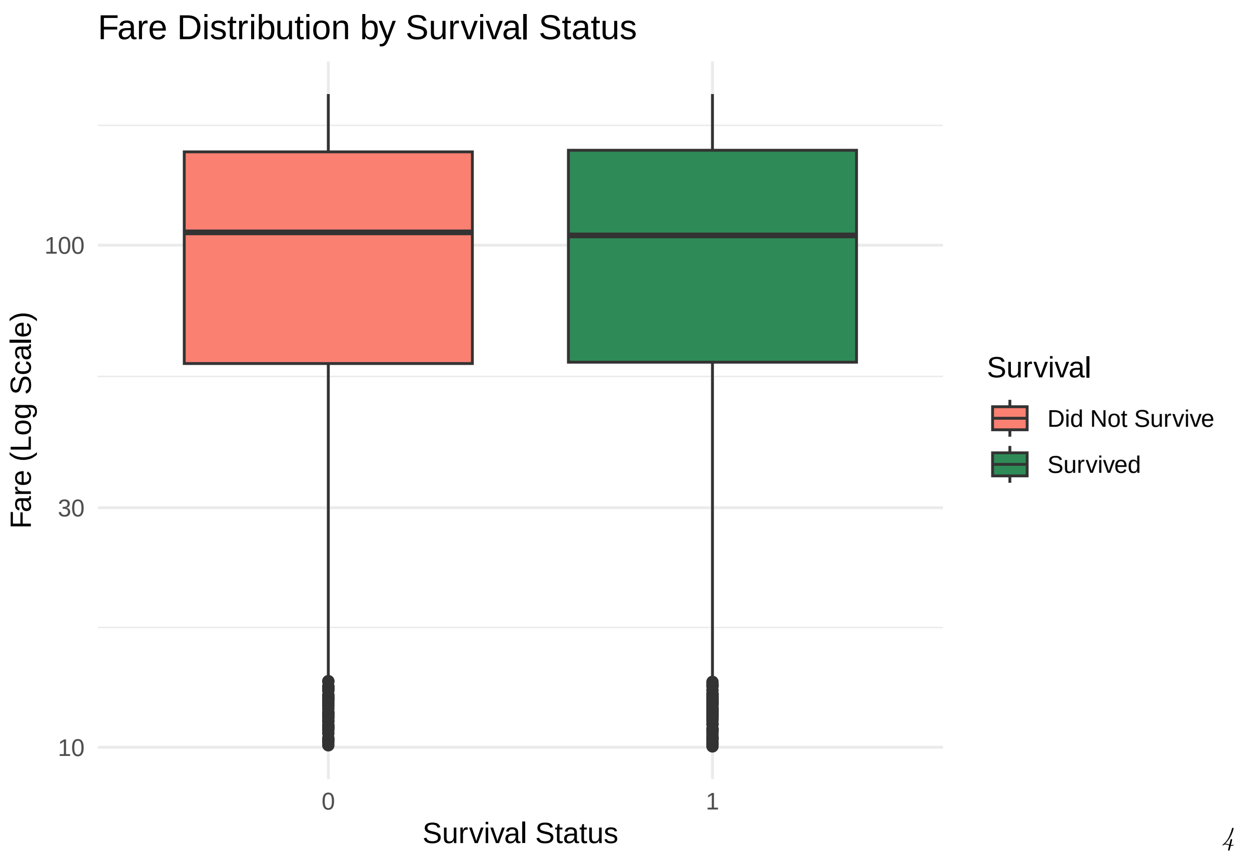
```r
ggplot(df, aes(x = Embarked, fill = Embarked)) +
  geom_bar() +
  labs(title = "Distribution of Embarkation Ports",
       x = "Port of Embarkation",
       y = "Number of Passengers") +
  theme_minimal()
```



4.8

Relationship between Fare and Survival (Boxplots) This box plot examines if there's a relationship between the fare paid and survival status. A logarithmic scale is applied to the Fare axis for better visualization of its distribution.

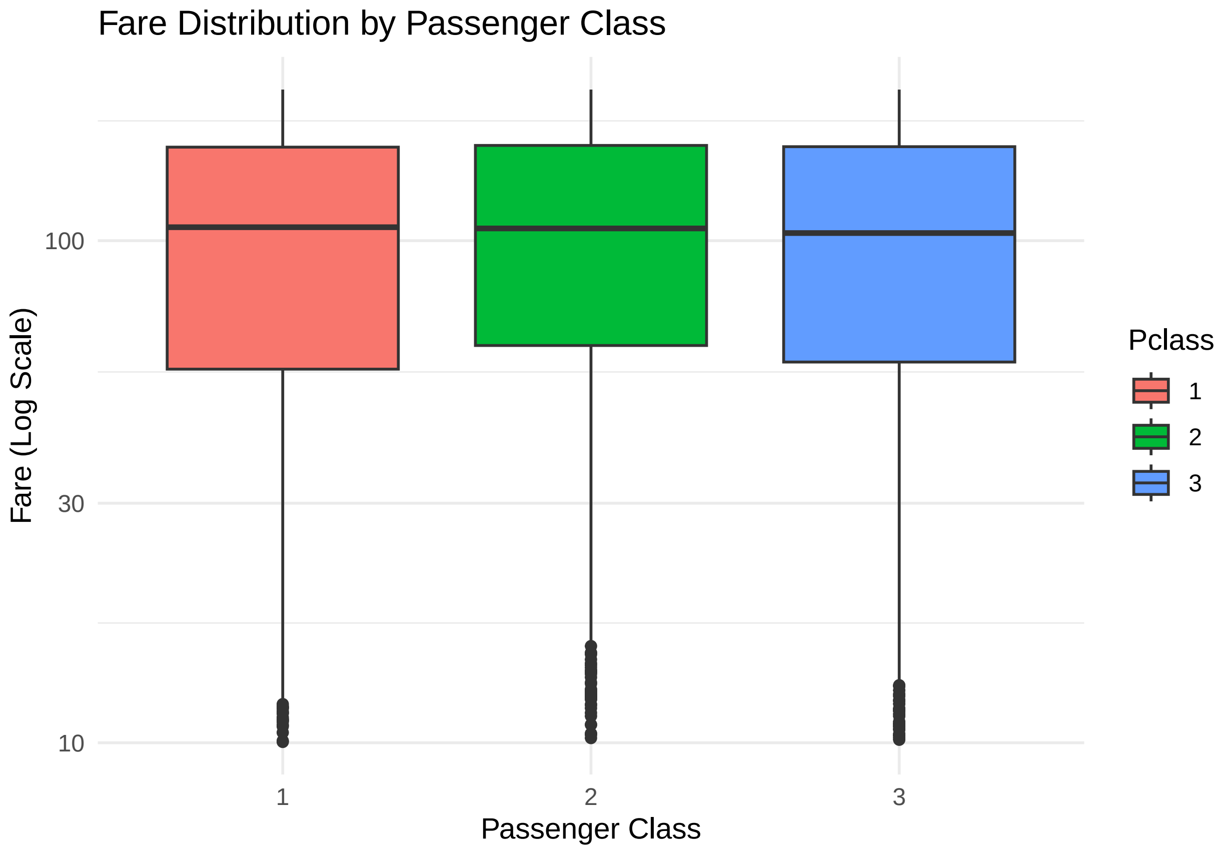
```r
ggplot(df, aes(x = Survived, y = Fare, fill = Survived)) +
  geom_boxplot() +
  scale_fill_manual(values = c("0" = "salmon", "1" = "seagreen"),
                    labels = c("Did Not Survive", "Survived")) +
  scale_y_log10() + # Use log scale for Fare
  labs(title = "Fare Distribution by Survival Status",
       x = "Survival Status",
       y = "Fare (Log Scale)",
       fill = "Survival") +
  theme_minimal()
```



4.9

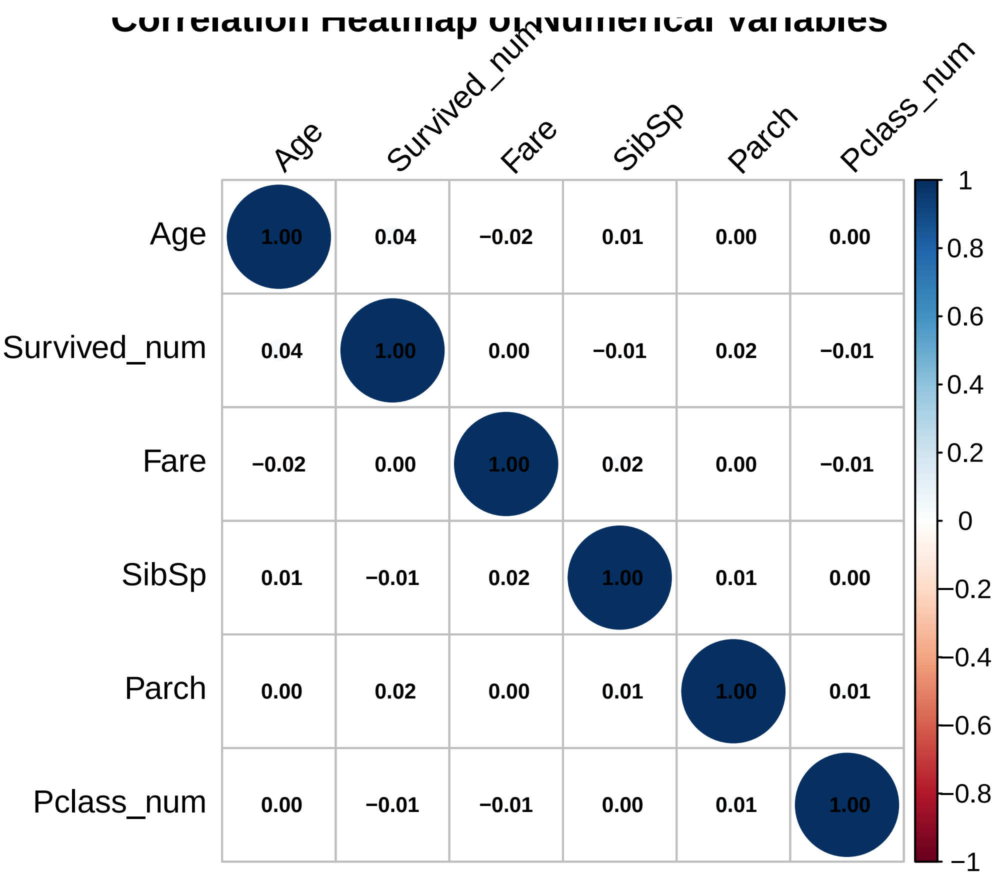
Fare Distribution Across Passenger Classes (Boxplots) This box plot visualizes the distribution of ticket fares across different passenger classes. A logarithmic scale is used for Fare due to its skewed nature, making the distribution clearer.

```
ggplot(df, aes(x = Pclass, y = Fare, fill = Pclass)) +
  geom_boxplot() +
  scale_y_log10() + # Use log scale for Fare due to its skewed distribution
  labs(title = "Fare Distribution by Passenger Class",
       x = "Passenger Class",
       y = "Fare (Log Scale)") +
  theme_minimal()
```



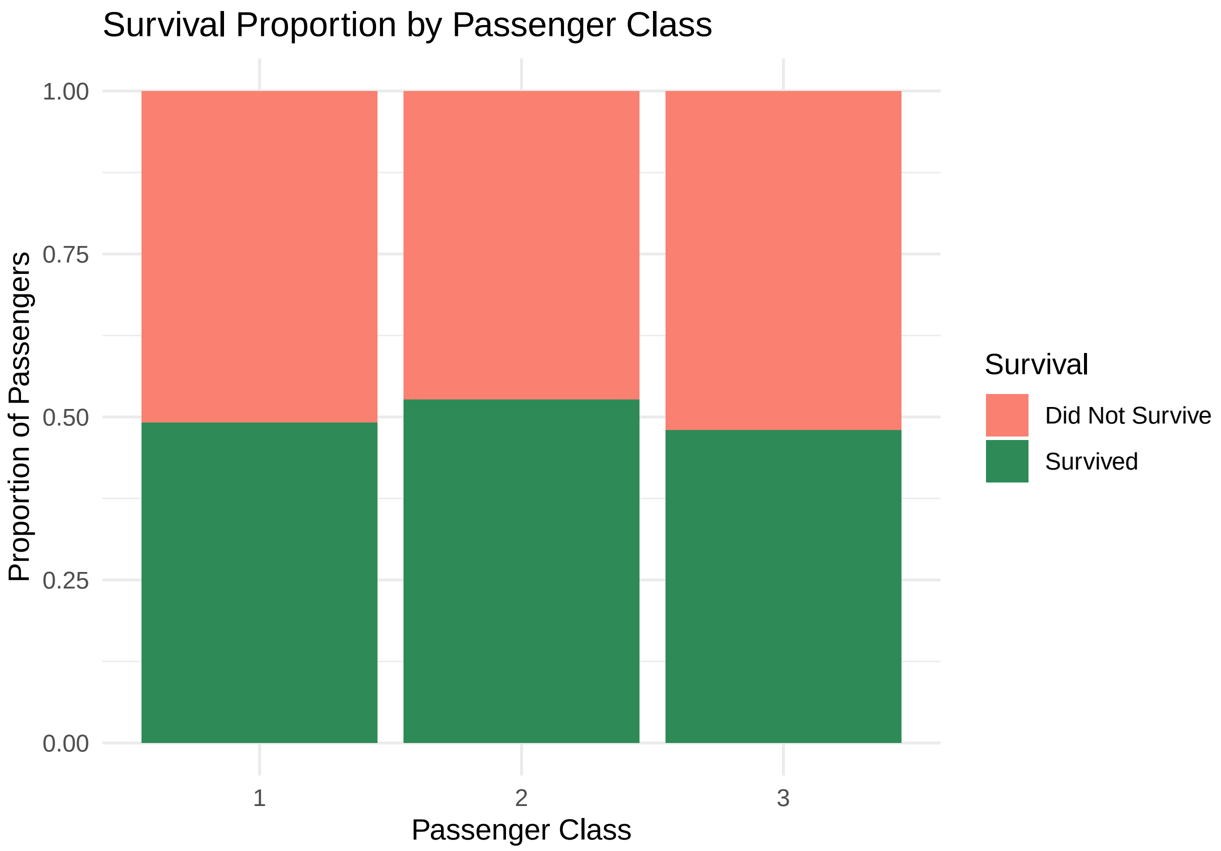
4.10 Correlation Heatmap for Numerical Variables (corrplot) A correlation heatmap visually represents the correlation coefficients between numerical variables. Stronger colors and larger circles indicate stronger correlations (positive or negative). This helps identify which numerical features move together.

```
numerical_df <- df %>%
  select(Age, Fare, SibSp, Parch) %>%
  mutate(Pclass_num = as.numeric(as.character(df$Pclass)),
         Survived_num = as.numeric(as.character(df$Survived)))

correlation_matrix <- cor(numerical_df, use = "pairwise.complete.obs") # Handle any remaining NAs if an

corrplot(correlation_matrix, method = "circle", type = "full",
         order = "hclust", tl.col = "black", tl.srt = 45,
         addCoef.col = "black", # Add correlation coefficients
         number.cex = 0.7,       # Size of coefficients
         main = "Correlation Heatmap of Numerical Variables")
```



4.11 Stacked Bar Chart: Survival by Passenger Class This chart displays the proportion of survivors and non-survivors within each passenger class, allowing for a direct comparison of survival rates across classes.

```r
ggplot(df, aes(x = Pclass, fill = Survived)) +
  geom_bar(position = "fill") + # "fill" for stacked percentages
  scale_fill_manual(values = c("0" = "salmon", "1" = "seagreen"),
                    labels = c("Did Not Survive", "Survived")) +
  labs(title = "Survival Proportion by Passenger Class",
       x = "Passenger Class",
       y = "Proportion of Passengers",
       fill = "Survival") +
  theme_minimal()
```



4.12 Most Frequent Cabin Entries (excluding missing) This code snippet identifies and displays the top 10 most frequently assigned cabin numbers, excluding any missing values. This can provide insights into common cabin assignments.

```r
# Drop NAs and get value counts
cabin_counts <- table(df$Cabin[!is.na(df$Cabin)])
# Display top 10 most frequent cabins
head(sort(cabin_counts, decreasing = TRUE), 10)
```

```
##
##       B57  E44  D33 C123   G6
##  859  842  838  827  821  813
```

5. Exporting Cleaned Data from R for Tableau After completing the data cleaning, transformation, and initial analysis in R, the final step in this phase is to export the modified dataset. This clean and enriched dataset will then serve as the primary data source for building an interactive dashboard in Tableau.

```r
write.csv(df, "titanic_cleaned_for_tableau.csv", row.names = FALSE)
```